

Javadoc generation

PDFDoclet

Doclet Name

com.tarsec.javadoc.pdfdoclet.PDFDoclet

Doclet Classpath

D:\aurigadoclet\bin\AurigaDoclet.jar

VM Options

-J-Xmx512m

Extra Javadoc options

Minimum configuration

-pdf C:\Users\felix.vincent\Desktop\PDFDoclet_LLD.pdf

-author

-version

-summary.table

-create.links

-allow.printing

-create.frame

Configuration

Overview

PDFDoclet should be configured through a property file. This is (IMHO) the clearest and easiest way of maintaining a set of configuration values (instead of using dozens of parameters). However, all settings can also be configured through commandline parameters.

For using a property file, specify the file with the "-config" commandline argument:

```
-config <property file path>
```

All other configuration values can be specified as an argument or in a property file. For example, to print the "author" tag, either the commandline argument "-author" may be specified, or the property "author" may be set in the config file.

Note that all properties with a boolean value ("true"/"yes" or "false"/"no") DON'T TAKE A VALUE when specified as commandline argument. In that case, they are by default assumed to be "true". For example, specifying "-author" (without a specific value) on the commandline has the same meaning as specifying "author=true" or "author=yes" in the configuration file.

Properties

Here is a list of all available configuration properties. The list has been split in two parts: The first lists all simple boolean "yes"/"no" properties. The second list shows all the properties which need an additional value argument on the command line.

Boolean properties (value "yes"/"no")

| Property | Description |
|----------------|--|
| debug | Lets the doclet print additional output in the console and to a logfile. |
| author | Print "Author" tags |
| version | Print "Version" tags |
| tag.since | Print "since" tags |
| summary.table | Create summary tables |
| create.links | Create hyperlinks |
| encrypted | Encrypt the PDF file |
| allow.printing | Allow printing of the PDF file |
| create.frame | Create a bookmark frame |
| api.title.page | Print a title page |
| filter | Enables the tag-based filter |

Properties with a value

| Property | Description |
|-----------------|--|
| pdf <file> | The path of the created PDF file. |
| workdir <dir> | The base directory for image file path specifications (default is the current directory). |
| sort <list> | Optional comma-separated list of package names to define the ordering of the packages in the bookmarks frame. |
| dontspec <list> | Optional comma-separated list of package names to define packages whose classes should not be printed fully qualified (just "String" instead of "java.lang.String"). |

| | |
|--|---|
| <code>label.tag.<name> <label></code> | Specifies a custom doc tag with a certain label. |
| <code>label.bookmarks.title</code> | Overrides the text label to be printed for the title of the bookmarks. |
| <code>label.bookmarks.index</code> | Overrides the text label to be printed for the index entry in the bookmarks. |
| <code>label.bookmarks.overview</code> | Overrides the text label to be printed for the overview entry in the bookmarks. |
| <code>label.bookmarks.packages</code> | Overrides the text label to be printed for the packages entry in the bookmarks. |
| <code>label.bookmarks.otherPackages</code> | Only used with the <code>-group</code> parameter. Overrides the text label to be printed for the "Other Packages" entry in the bookmarks. |
| <code>label.fields</code> | Overrides the text label to be printed for the fields entry in the bookmarks. |
| <code>label.constructors</code> | Overrides the text label to be printed for the constructors entry in the bookmarks. |
| <code>label.methods</code> | Overrides the text label to be printed for the methods entry in the bookmarks. |
| <code>label.tag.<tagname></code> | Defines the text label for a tag. Can either override the label of a default tag such as "author", or define the label for a custom tag (custom tags are not printed as long as no label is defined for them). |
| <code>inherited.summary.table=[yes internal]</code> | Create summary tables for inherited fields/methods. If the value is set to <code>internal</code> , summary tables are printed only for elements inherited by classes and interfaces within the scope of the API. Elements inherited from other classes such as <code>java.lang.Object</code> are ignored. |
| <code>appendix.<number>.name</code> | Defines the name / number of the appendix. |
| <code>appendix.<number>.title</code> | Defines the title label of the appendix. |
| <code>appendix.<number>.file=<file path>[,page,from-to,...]</code> | Path of the HTML or PDF file that should be included. If a PDF file is specified, the filename may be followed by a comma-separated list of single page numbers or page ranges, to define which pages to import from the PDF |

| | |
|-------------------------------|---|
| | document. |
| page.numbers.prefix <text> | Prefix-text for page numbers. |
| page.numbers.type <type> | "full" or "simple" |
| page.numbers.alignment <type> | "left", "center", "right" or "switch". Defines the alignment of page numbers. "switch" means that they will switch from left to right and back, as needed for printing. |
| api.title.file <file> | The path of a HTML file used for the title page of the PDF. |
| api.title <file> | Title for the title page if no external HTML title page is used. |
| api.copyright <copyright> | Copyright remark for the title page if no external HTML title page is used. |
| api.author <author> | Author's name for the title page if no external HTML title page is used. |
| font.text.name <ttf-file> | Font to be used for all regular text. Default is TIMES-ROMAN (path of TrueType-font, like ./fonts/garait.ttf). |
| font.text.enc | Optional encoding for the text font. |
| font.code.name <ttf-file> | Font to be used for all fixed-width code parts. Default is COURIER. |
| font.code.enc | Optional encoding for the code font. |
| filter.tags <taglist> | Comma-separated list of filter tags. |
| filter.tag.<name> <label> | Filter tag with certain name and value. |

Font embedding

As for the fonts: The specified font will be embedded into the PDF. If that is not possible (due to legal restrictions of the font), the doclet will fail to create the PDF.

Overriding properties with parameters

Please note: A configuration file can be used in combination with command-line arguments. However, every configuration value specified on the command line will override the same configuration property from the property file. For example, if the command-line parameter "-workdir=./example/src3" is used, but in the configuration file there's a property "workdir=./example/src1", the value will be "./example/src3".

IMPORTANT NOTE: In versions 0.6.0 and older of PDFDoclet, it was exactly the other way around, which was not very logical and intuitive. Therefore I decided to change it (better getting rid of bad implementation at the risk of causing some temporary confusion than keeping it forever).

For information about how to use additional parameters with a custom doclet, read Sun's javadoc instructions. Also check out the targets "example" and "test-pdf" in the build.xml file to see how it can be done.

AurigaDoclet

Doclet Name

com.aurigalogic.doclet.core.Doclet

Doclet Classpath

D:\aurigadoclet\bin\AurigaDoclet.jar

VM Options

-J-Xmx512m

Extra Javadoc options

Minimum configuration

-format pdf

-out C:\Users\felix.vincent\Desktop\AurigaLLD.pdf

Others

-format

The output format.

Supported values: fo,pdf,ps,pcl,svg.

-out

Output file path.

-notoc

Do not generate TOC page.

-nonavigation

Do not generate navigation tree.

-nolinks

Do not use hyperlinks.

-noindex

Do not generate a keyword index.

-leftmargin

Left margin in points. Default is 30.

-rightmargin

Right margin in points. Default is 30.

-topmargin

Top margin in points. Default is 10.

-bottommargin

Bottom margin in points. Default is 10.

-headertext

XHTML text to be used as page header.

-headerfile

XHTML file to be used as page header.

-footertext

XHTML text to be used as page footer.

-footerfile

XHTML file to be used as page footer.

-headerheight

Height of page headers in points. Default is 50.

-footerheight

Height of page footer in points. Default is 20.

-coverfile

XHTML file to be used a cover page.

-cssfile

CSS file to used for formatting the output.

Default css file is located in *src/com/aurigalogic/doclet/resources/default.css*

-xslfile

Custom xsl file to be used for formatting the output.

Default xsl is located in *src/com/aurigalogic/doclet/resources/xml2fo.xsl*

DocFlex/Javadoc

Doclet Name

com.docflex.javadoc.Doclet

Doclet Classpath

D:\docflex-doclet-1.6.1\lib\docflex-doclet.jar

VM Options

-J-Xmx512m

Extra Javadoc options

Minimum configuration

-template D:\docflex-doclet-1.6.1\templates\javadoc\PlainDoc.tpl

-format RTF

-d C:\Users\felix.vincent\Desktop

-f DocFlex_LLD

Command Line Options

Here are all command line options specifically processed by [DocFlex Doclet](#):

| | | |
|--------------------------------|-------------------------------|---------------------------|
| -config | -format | -nodialog |
| -d | -launchviewer | -o |
| -docflexconfig | -license | -p |
| -errlog | -link | -quiet |
| -f | -linkoffline | -template |

The number of options is relatively small for two reasons:

1. Since all informative content of the generated documentation is programmed entirely within templates, it is the matter of template parameters to control this. The templates parameters are passed via the [-p option](#).
2. The output formats do have their specific options. But since such options are different for each particular output format and the number of supported formats will grow, all output format options are specified in the uniform way via [-o option](#) (the same as the template parameters).

Effectively, the number of various generator settings may be so great that the [DocFlex Doclet](#) provides a different and more convenient way for specifying all of them -- the [Doclet GUI](#).

Here are the option details grouped by category:

Template Options

-template <file>

Specifies the main template file to be executed by the generator. All documentation generation is controlled by this template. The main template may call from itself another templates (*subtemplates*) producing either a single output file or multiple files (framed documentation).

-p:<parameter>=<value>

or

-p:<parameter> <value>

Specifies the value of a template parameter.

Notes:

- You can find the necessary parameter names in the [Template Parameter Inspector](#) invoked from the [Generator Dialog](#).
- The boolean values should be specified as "true" and "false" strings (for example: `-p:include.deprecated=true`).
- There are also *enum* parameters, which are strings limited to only fixed number of values. You can see all possible values accepted by an enum parameter in [Parameter Inspector](#).
- Now DocFlex supports [multivalued \(list\) parameters](#), which can be equally specified on the command line using this option. See [Specifying List Parameter on Command Line](#) for more details.
- When the parameter value contains spaces, enclose it in double quotes (for example: `-p:title "My Docs"`).

- The first variant of this option allows you to specify the parameter with a single command line argument, the second variant breaks the option into two arguments (one for the parameter name, another one for the parameter value).

The two argument variant is necessary for compatibility with [Apache Ant](#). Ant provides a special doclet `<param>` element with two attributes: `name` and `value`. When everything goes into the `name` attribute and the actual value contains spaces, Ant won't process it correctly.

Output Options

-format `<HTML | RTF | TXT>`

Specifies the output format. By default, HTML.

-o:`<option>=<value>`

or

-o:`<option> <value>`

Specifies the output format option.

Notes:

- You can find the necessary option names in the [Format Option Inspector](#) invoked from the [Generator Dialog](#).
- The values of the boolean options should be specified as "true" and "false" strings (for example: `-o:rtf.storeGraphicsInRTF=false`).
- The first variant of this option allows you to specify everything with a single command line argument, the second variant breaks the option into two arguments (one for the option name, another one for the value).

The two argument variant is necessary for compatibility with [Apache Ant](#). Ant provides a special doclet `<param>` element with two attributes: `name` and `value`. When everything goes into the `name` attribute and the actual option value contains spaces, Ant won't process it correctly.

-d `<directory>`

Specifies the destination directory for the generated documentation. By default, this is the current directory (the one designated by "." pathname). This option works in conjunction with the [-f option](#).

The documentation files are distributed in the following way:

- **In the case of a single-file documentation**
 1. The output document is placed in the destination directory under default file name '*templatename.formatextention*' (for example, in the RTF output format, template PlainDoc.tpl will produce document 'PlainDoc.rtf'). The [-f option](#) may override this name.

2. All associated files (such as images, if not included in the document) are placed in the '*docname_files*' subdirectory near the main document.
- **In the case of a framed multiple-file documentation**
 1. By default, all generated files and subdirectories are placed in the specified destination directory. The frameset file produced by the main template is saved under the name 'index.html'. This is the same as the output produced by the [Standard Javadoc doclet](#).
 2. If a different name '*docname*' is specified with the [-f option](#), the frameset file is saved under this name. All other files and the subdirectory tree are moved into '*docname_files*' subdirectory near the frameset file.

This makes the whole documentation to look as to consist of only two file entities, which may be easier to distribute yet during the generation, especially when different types of documentation are produced from the same Java sources (e.g. by different templates or with the different parameters).

Examples:

```
-template PlainDoc.tpl -format RTF -d c:\out
```

generate the document file `c:\out\PlainDoc.rtf` with the associated files subdirectory `c:\out\PlainDoc_files\` (if any)

```
-template PlainDoc.tpl -format HTML -d c:\out -f OpenAPI
```

generate the document file `c:\out\OpenAPI.html` with the associated files subdirectory `c:\out\OpenAPI_files\` (if any)

```
-template FramedDoc.tpl -format HTML -d c:\out
```

generate the framed documentation located in the directory `c:\out\` with the main file `index.html`

```
-template FramedDoc.tpl -format HTML -d c:\out -f OpenAPI
```

generate the framed documentation with the main file `c:\out\OpenAPI.html` and all other files located in the directory `c:\out\OpenAPI_files\`

-f <file>

Specifies the documentation output file name. This option works in conjunction with the [-d option](#) and specifies the name of the main output file (the one associated with the main template).

Typically, it should be used to specify a pure name associated with the generated documentation (for example, 'DocletAPI'). However, the pathname may also be used. In that case, it is interpreted against the initial destination directory (specified with the **-d** option) and may override it. If the pathname is the absolute one, the **-d** option is effectively ignored.

-launchviewer[=<true / false>]

Tells the generator to execute a specific non-Java command to launch an external application able to view the generated result (for instance, an Internet browser to view HTML documentation).

Precisely, this command is specified in [docflex.config](#) file found near `docflex-javadoc.jar` (or `docflex-doclet.jar`) file in the `lib` directory.

Examples:

-launchviewer

Do launch viewer

-launchviewer=false

Do not launch viewer, no matter what's specified in the [generator.config](#)

Configuration Options

-config <file>

Specifies the [generator config file](#), which may contain options written manually or using the [Generator Dialog](#). All options not provided directly on the command line are looked for in this file.

The Generator Dialog also uses this file to store all options and parameters specified interactively. If the config file does not exist, the Generator Dialog will try to create it by the provided pathname.

By default, the location of the generator configuration file is specified in the [docflex.config](#) (located in `lib` directory). If no other config file is provided, it is automatically loaded from that location (if found).

See also: [Handling Template Parameters | Using Doclet GUI to prepare parameters for command line](#)

-docflexconfig <file>

Specifies an alternative path to the DocFlex [main configuration file](#). By default, this file is 'docflex.config' which is searched in the directory where the DocFlex Java library file 'docflex-javadoc.jar' (or `docflex-doclet.jar`) is located. If not found, [docflex.config](#) is created automatically with default settings.

-license <file>

Specifies the location of the [license file](#).

You may need this option when you use the [full edition](#) of DocFlex/Javadoc and your license allows you to execute your templates only in its presence. At that, the way how the license is searched by default does not fit to you (see [Documentation | Configuration Files | docflex.license](#)).

Note: The license directly specified with this option on the command line will override any other licenses (e.g. assigned in [docflex.config](#) or found near `docflex-javadoc.jar`).

Other Options

-link <extdocURL>

Creates links to existing javadoc-generated documentation of external referenced classes.

This option will work the same as the [-link](#) option supported by the [Standard Doclet](#).

Precisely, the processing of both `-link` and [-linkoffline](#) options is programmed within the templates using a Javadoc-specific [FlexQuery](#) function: `getExternalDocURL()`. When DocFlex/Doclet meets either `-link` or `-linkoffline` option on the Javadoc command line, it processes such an option so that the content of the `package-list` file referred from it is loaded into a special internal hash-table. Further, `getExternalDocURL()` function uses that table to translate any Java qualified name into a possible URL of the external documentation.

-linkoffline <extdocURL> <packagelistLoc>

This option is a variation of [-link](#); they both create links to javadoc-generated documentation for external referenced classes. Use the `-linkoffline` option when linking to a document on the web when the Javadoc tool itself is "offline" -- that is, it cannot access the document through a web connection.

This option will work the same as the [-linkoffline](#) option provided by the [Standard Doclet](#). See also [-link](#) option for more details about how it is implemented.

-nodialog[=<true / [false](#) / *quiet*>]

Do not invoke the [Doclet GUI](#).

Parameters:

true -if this (or no parameter) is specified, the [DocFlex Doclet](#) starts the generation immediately according to the setting specified on the Javadoc command line and loaded from the [generator config](#) (see [-config](#) option). Then, the doclet exits.

false -this is the same as when the `-nodialog` option was not specified at all. The [Generator Dialog](#) will be invoked.

quiet -start the generation immediately (without dialog) and suppress displaying most of the generator messages to screen (see also [-quiet](#) option). The generation progress counter will be printed.

-errlog <file>

Specifies the **error log file** used when the [DocFlex Doclet](#) is executed without the [Generator Dialog](#) (i.e. when `-nodialog` option is also specified on the command line).

By default, when an unexpected error/exception occurs during the generation and no GUI is enabled, all details about the error are printed to the standard console.

Using this option, you can assign a separate error log file, into which the detailed ERROR REPORT is dumped each time an error happens. Only brief messages will get on the console in that case. (See also [Error Reporting](#) for more details.)

The error log file should be specified as an absolute or relative file pathname. When the pathname points to a directory, it will be extended with the default `"docflex_error.log"` name (for example, setting `"-errlog ."` will be interpreted as `'docflex_error.log'` file located in the current directory).

If the error log file does not exist, it is created in the event of error. Otherwise, the ERROR REPORT is appended to the existing file. In the case of any I/O error related to the error log file itself, everything will be printed to the console (along with additional the log file error message).

-quiet[=*true* / *false*>]

Suppress displaying most of the generator messages to screen.

Note: Since Java 5, this option is used by Javadoc itself as one of its core options. If you want to suppress only [DocFlex Doclet](#) messages but still see the generation progress counter, use `-nodialog=quiet` option instead.